--Summary of the Invention--;

Page 3, at line 9 and before the paragraph beginning "Other characteristics...", insert the following heading at the left hand margin:

--Brief Description of the Drawings--;

Page 3, at line 17, before the paragraph beginning "Fig. 1...", insert the following heading at the left-hand margin:

--Description of the Preferred Embodiments--;

Page 6, line 25, after "table", delete "7" and substitute –(7)--;

IN THE CLAIMS:

Please cancel claims 1- 7 in their entirety and without prejudice and substitute the following new claims:

--8. A high-performance specification resolution method for use in detecting attacks against computer systems comprising:

a) formulating audit conditions to be detected using non-limiting specification formulas expressing fraudulent entry or attack patterns or abnormal operations, to be verified by examining the records of a log file of the computer system;

b) expanding said formulas into subformulas;

c) scanning by an interpreter, and generating, for each expanded formula in each record, Horn clauses to resolve in order to detect whether or not the formula is valid in the record, the Horn clauses expressing the implications resolvent of the subformulas for each record scanned, in positive clauses, i.e. counting only a positive literal and in non-positive clauses, i.e. counting at least one negative literal, which negative literals form the negative part of the clause;

d) storing positive Horn clauses in a stack of worked subformulas, and storing, in a table comprising a representation, implicating subformula(s) constituting the negative part of the clause and the link with the implicated subformula(s) constituting the positive part of the clause, and storing in a counter the number of formulas or subformulas present in the negative part of the clause for each implicated subformula;

e) resolving the table based on each positive clause encountered, so as to generate either an output file or an action of the computer system;

f) iterating steps b) through e) until the scanning of all the records in the log file is complete.

9 A method according to claim 8, characterized in that a temporal logic is used for the formulation of the specification.

10. A method according to claim 8, characterized in that the table is a matrix and is indexed in columns by subscripts of the formulas appearing in the negative part of the Horn clauses, and the lines are the Horn clauses exactly.

11. A method according to claim 8, characterized in that the table is preferably represented in the form of a sparse matrix, the columns being represented by means of chained lists and the implicit lines.

12. A method according to claim 8, characterized in that a step for optimizing the expansion of the formulas is obtained through a hash table to ensure that the same formula is not expanded more than once in each record.

13. A method according to claim 9, characterized in that a step for optimizing the expansion of the formulas is obtained through a hash table to ensure that the same formula is not expanded more than once in each record.

14. A method according to claim 8, characterized in that the log file is scanned only once from beginning to end.

15. A computer system comprising storage means and means for executing programs for implementing a high performance resolution method for deleting attacks against the system wherein the method:

a) formulates audit conditions to be detected using non-limiting specification formulas expressing fraudulent entry or attack patterns or abnormal operations, to be verified by examining the records of a log file of the computer system;

b) expands said formulas into subformulas;

c) scans by an interpreter, and generates, for each expanded formula in each record, Horn clauses to resolve in order to detect whether or not the formula is valid in the record, the Horn clauses expressing the implications resolvent of the subformulas for each record scanned, in positive clauses, i.e. counting only a positive literal and in non-positive clauses, i.e. counting at least one negative literal, which negative literals form the negative part of the clause;

d) stores positive Horn clauses in a stack of worked subformulas, and storing, in a table comprising a representation, implicating subformula(s) constituting the negative part of the clause and the link with the implicated subformula(s) constituting the positive part of the clause, and stores in a counter the number of formulas or subformulas present in the negative part of the clause for each implicated subformula; and

e) resolves the table based on each positive clause encountered, so as to generate either an output file or an action of the computer system;

- an adaptor for translating information from a log file formulated in the specific language of the machine into a language comprehensible to an interpreter;

- the interpreter receiving the information from the adapter and receiving the formulation of the specification in a temporal logic in a specification formula in order to expand said formula and fill in the table and the stack of worked subformulas stored in a memory of the computer system and resulting from the scanning of the computer system's log file;

- a clause processing algorithm executed by the computer system, for resolving the Horn clauses using the information from the table and the stack of worked subformulas, said clause processing algorithm generating an output file or generating an action.

16. A computer system as defined in claim 15 wherein the temporal logic is used for formulation of the specification.

17. A computer system as defined in claim 15, wherein the table is a matrix and is indexed in columns by subscripts of the formulas appearing in the negative part of the Horn clauses, and the lines are the Horn clauses exactly.

18. A computer system as defined in claim 15, wherein the table is preferably represented in the form of a sparse matrix, the columns being represented by means of chained lists and the implicit lines.

19. A computer system as defined in claim 15 including a hash table to ensure that the same formula is not expanded more than once in each record.

20. A computer system as defined in claim 16 including a hash table to ensure that the same formula is not expanded more than once in each record.

21. A computer system as defined in claim 15 including means for scanning the log file only once from beginning to end.--